



THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Lagged correlation-based deep learning for directional trend change prediction in financial time series

**Citation for published version:**

Moews, B, Herrmann, JM & Ibikunle, G 2019, 'Lagged correlation-based deep learning for directional trend change prediction in financial time series', *Expert Systems with Applications*, vol. 120, pp. 197-206.  
<https://doi.org/10.1016/j.eswa.2018.11.027>

**Digital Object Identifier (DOI):**

[10.1016/j.eswa.2018.11.027](https://doi.org/10.1016/j.eswa.2018.11.027)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Peer reviewed version

**Published In:**

Expert Systems with Applications

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Lagged correlation-based deep learning for directional trend change prediction in financial time series

Ben Moews<sup>a,\*</sup>, J. Michael Herrmann<sup>b</sup>, Gbenga Ibikunle<sup>c</sup>

<sup>a</sup>*School of Physics & Astronomy, University of Edinburgh, Edinburgh, EH9 3HJ, UK*

<sup>b</sup>*School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK*

<sup>c</sup>*Business School, University of Edinburgh, Edinburgh, EH9 3HJ, UK*

---

## Abstract

Trend change prediction in complex systems with a large number of noisy time series is a problem with many applications for real-world phenomena, with stock markets as a notoriously difficult to predict example of such systems. We approach predictions of directional trend changes via complex lagged correlations between them, excluding any information about the target series from the respective inputs to achieve predictions purely based on such correlations with other series. We propose the use of deep neural networks that employ step-wise linear regressions with exponential smoothing in the preparatory feature engineering for this task, with regression slopes as trend strength indicators for a given time interval. We apply this method to historical stock market data from 2011 to 2016 as a use case example of lagged correlations between large numbers of time series that are heavily influenced by externally arising new information as a random factor. The results demonstrate the viability of the proposed approach, with state-of-the-art accuracies and accounting for the statistical significance of the results for additional validation, as well as important implications for modern financial economics.

*Keywords:* Lagged correlation, Deep learning, Trend analysis, Stock markets  
*2010 MSC:* 68T05, 62P20

---

## 1. Introduction

An increased interest in deep-layered machine learning approaches for time series analysis and forecasting resulted in applications in various fields, establishing this area as a challenging topic of interest (Cao and Tay, 2003; Nesreen et al., 2010). When it comes to the effective use of deep neural networks, one of the primary concerns is a sensible approach to feature engineering for useful data representations. This process often depends on domain knowledge about

---

\*Corresponding author

*Email addresses:* `b.moews@ed.ac.uk` (Ben Moews), `michael.herrmann@ed.ac.uk` (J. Michael Herrmann), `gbenga.ibikunle@ed.ac.uk` (Gbenga Ibikunle)

the respective area of application and is, more often than not, a time-consuming part of research (Najafabadi et al., 2015). Some researchers equate machine learning, in an attempt to emphasize the relative importance, with the concept of feature engineering itself (Ng, 2012). Such representations have to be informationally rich enough to incorporate the looked-for lagged correlations between time series while, at the same time, being constrained to a discrete number per observation and variable for input features in a feed-forward neural network. Zhang and Qi (2005) find that feed-forward neural networks are not able to capture the necessary information when applied to raw data from time series with seasonal and trend patterns, which opens the field for approaches to feature engineering that allow for an effective use of time series data for trend predictions in a variety of application areas.

In this paper, we test the hypothesis that deep feed-forward neural networks combined with exponential smoothing for the training inputs are suitable for learning lagged correlations between the step-wise trends of a large number of time series, and that such models can be successfully applied to current research on real-world forecasting problems. In order to test this approach, we apply the proposed method to gradients computed for five years of historical stock price data of the S&P 500 stocks in one-hour intervals for daily trends, adding the complication of relatively few observations. For a more in-depth overview of soft computing methods in financial market research, interested readers are referred to Cavalcante et al. (2016), with Weng et al. (2018) providing an application of ensemble methods to financial markets using a variety of text-based and index-based features.

The experiments that are conducted for this purpose demonstrate the viability of this approach by predicting price trend changes with an accuracy above given market baselines and within a stringent statistical validation framework. In order to evaluate the soundness of our conclusions, we test the results against the alternative possibilities of simply learning frequencies or probabilistic distributions, calculate confidence intervals and  $p$ -values, as well as a visual analysis via notched box plots (McGill et al., 1978). The results of this paper deliver evidence for the applicability to a number of real-world problems that deal with complex relationships of large numbers of noisy time series.

The results for the specific tests performed for this paper are also discussed in relation to the random walk hypothesis and the efficient-market hypothesis in financial economics.

Our hypothesis relates to the microstructure model by Ho and Stoll (1983), which characterizes the links between quote changes in a stock and the evolution of the inventory with respect to the other stocks. The model shows that quote changes in stock  $a$ , which is in reaction to a transaction in stock  $b$ , are based on  $\text{cov}(Ra, Rb)/\sigma^2(Rb)$ . This portfolio view of stock trading is in line with commonly deployed diversification strategies in finance and has given rise to highly popular instruments such as exchange traded funds (ETFs), which offer cheap means of diversifying risk. This study has implications for perhaps the longest running debate in the financial economics literature.

The unpredictability of the factors influencing price discovery in stocks

54 makes the price discovery process noisy (Chen et al., 1986). The unpredictabil-  
55 ity (or randomness) of the information acquisition process in financial markets  
56 is consistent with the efficient market hypothesis described by Fama (1965) and  
57 Fama (1970) as well as the random walk hypothesis (Kendall and Hill, 1953;  
58 Cootner, 1964; Malkiel, 1973). These theories contradict our hypothesis on the  
59 existence of time-shifted correlations in stock markets. Consistent with our ex-  
60 pectations, our results deliver rigorously tested empirical evidence, supporting  
61 the existence of time-shifted correlations in stock prices and thus contradict the  
62 random walk theory. Specifically, our findings are inconsistent with Sitte and  
63 Sitte (2002), who argue that the price discovery process for S&P 500 stocks is  
64 a random walk due to the inability of artificial neural networks to extract any  
65 information resulting in above-average predictions for those stocks.

66 Our results are, however, consistent with previous, albeit weak, evidence for  
67 the absence of a random walk in financial time series via the use of artificial  
68 neural networks as presented by Darrat and Zhong (2000). The consistency of  
69 the efficient market hypothesis with the random walk hypothesis also implies  
70 that our findings contradict much of the efficient market hypothesis, which is  
71 widely supported by a large section of the finance academic literature (Fama,  
72 1970; Doran et al., 2010). However, despite the seemingly established nature of  
73 the random walk hypothesis, over the years, many studies, like Lo and MacKin-  
74 lay (1987), have questioned its validity, while others have proposed alternatives.  
75 For example, one popular alternative hypothesis is that stock returns may be  
76 explained by the sum of a random walk and a stationary mean-reverting com-  
77 ponent (Summers, 1986; Fama and French, 1988). Lo and MacKinlay (1987)  
78 also advance the view that the efficient market hypothesis is an 'incomplete hy-  
79 pothesis'. This current paper is not an attempt to reconcile one of the longest  
80 standing debates in the finance literature; rather, we propose a price predic-  
81 tion approach based on an amalgamation of market microstructure theory and  
82 machine learning.

## 83 2. Related research

### 84 2.1. Trend prediction in time series

85 The feasibility of different types of artificial neural networks for trend pre-  
86 diction in time series was indicated early by Saad et al. (1998). Other types of  
87 noise reduction, for example PCA for echo state networks, have already been  
88 subjected to similar investigations, with no significant success being reported  
89 (Lin et al., 2009). While research on regression gradients as input features for  
90 feed-forward neural networks is sparse in the published literature, the usage of  
91 directional derivatives of wavelets was recently introduced in natural language  
92 processing (Gibson et al., 2013).

93 Features based on derivatives were subsequently adapted in other research  
94 areas, for example statistics and digital signal processing (Grecki and Łuczak,  
95 2013; Baggenstoss, 2015). The success of the regression derivative-based ap-  
96 proach for trend forecasting using lagged correlations between time series pre-  
97 sented here provides additional evidence for the viability of such methods for

time series applications. Specifically, positive results show the value and applicability of deep learning methods for such scenarios.

## 2.2. Stock markets as a use case

Price changes in stock markets are, at their core, the result of human decisions which, in turn, are based on their respective beliefs about stocks' future performance. Stocks are influenced not only by a company's respective performance, but also by newly arising information not directly linked to the latter. Examples are negative effects on stock prices for airlines after the 9/11 attacks, and similar effects after news about a CEO's diminishing personal health (Drakos, 2004; Perryman et al., 2010). Price changes are, therefore, the result of human beliefs about the future beliefs of other humans, which can be iterated indefinitely and is an example of real-world time series being created by human decision-making and the implementation of automated decision-making based on these notions, especially in high frequency trading.

It can be concluded that time series of historical stock prices contain, due to these factors, a large amount of noise in the form of new information influencing the process, and through biases and errors in human decision-making. Markets are, as a result, inherently prone to fluctuations triggered by overreactions, and to dynamical reinforcement during temporary crazes (Chen et al., 1986). This makes them, due to the complexity of their generation process, a suitable use case to test the proposed model's ability to identify and exploit lagged correlations in notoriously hard-to-predict noisy environments.

Relevant recent work on stock market prediction includes Zhang and Wu (2009) on changes to backpropagation, Boyacioglu and Avci (2010) on the use of fuzzy inference frameworks, Chatzis et al. (2018) on deep learning for financial crisis forecasting, Zhang et al. (2018) on unsupervised heuristic algorithms, and Malagrino et al. (2018) on Bayesian network approaches. Another area of research is concerned with text-based stock market prediction, with Nassirtoussi et al. (2014) providing a holistic overview for interested readers.

## 2.3. ANNs and stock markets

The viability of using artificial neural networks for stock market predictions was first hypothesized by White (1988), with subsequent indications of success by Saad et al. (1998) and Skabar and Cloete (2002). Zhang et al. (1997) report on the special suitability of artificial neural networks to such forecasting problems due to their adaptability, non-linearity and arbitrary function mapping. Takeuchi and Lee (2013) first made experimental results on the use of deep neural network models for stock market prediction available as a working paper by using the work of Hinton and Salakhutdinov (2006), and with a binary prediction accuracy of 53.36%. A similar approach by Batres-Estrada (2015) resulted in a comparable reported accuracy of 52.89%, also outperforming a simple logistic regression.

### 139 3. Gradients as features

#### 140 3.1. Approximating trend strengths

141 Linear regressions are a wide-spread approach to capture trends limited to a  
 142 certain time frame and, as such, form the basis for these features. They take, in  
 143 their general form, the following shape, with  $i$  indicating one of  $m$  observations,  
 144 intercept  $\beta_0$  and error term  $\epsilon_i$ :

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_p x_{i,p} + \epsilon_i \\ &= \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i, \quad i \in \{1, \dots, m\} \end{aligned} \quad (1)$$

145 As a one-variable feature is necessary for the model's input vector, a simple  
 146 linear regression is used as a least-squares estimator of equation (1). This es-  
 147 timator identifies one explanatory variable to minimize the squared sum of the  
 148 residuals by fitting a line. The equation is presented as a minimization problem  
 149 seeking  $\beta_0$  and the slope  $\beta_1$  for  $\min_{\beta_0, \beta_1} Q(\beta_0, \beta_1)$ :

$$Q(\beta_0, \beta_1) = \sum_{i=1}^m (y_i - \beta_0 - \beta_1 x_i)^2 \quad (2)$$

150 The gradient of the resulting regression model, that is, the slope of the fitted  
 151 line, can then be computed by taking the first derivative. In the given task, this  
 152 means extracting  $\beta_1$  as a feature. Information about the value of a time series  
 153 at a point along the timeline is lost in this process, with the resulting gradient  
 154 representing the strength of an upwards or downwards movement of a trend via  
 155 the regression model. A time interval size over which the regression is to be  
 156 performed has to be determined in advance. The feature matrix, which is later  
 157 used for the input of a feed-forward neural network, consists of the derivatives  
 158 w.r.t.  $\beta_{1,k}$  of a simple linear regression as in equation (2) per time series  $k \in$   
 159  $S := \{1, \dots, s\}$ , and for all separate time intervals  $j \in T := \{1, \dots, t\}$ ,  
 160 meaning that  $\forall j \in T, k \in S$ :

$$\begin{aligned} &\frac{\partial}{\partial \beta_{1,k}^j} \min_{\beta_{0,1}^j, \beta_{1,1}^j} \sum_{i=1}^m (y_{i,k} - \beta_{0,k}^j - \beta_{1,k}^j x_{i,k}^j)^2 \\ &\Rightarrow \begin{pmatrix} \beta_{1,1}^1 & \beta_{1,1}^2 & \cdot & \cdot & \cdot & \beta_{1,1}^t \\ \beta_{1,2}^1 & \beta_{1,2}^2 & \cdot & \cdot & \cdot & \beta_{1,2}^t \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \beta_{1,s}^1 & \beta_{1,s}^2 & \cdot & \cdot & \cdot & \beta_{1,s}^t \end{pmatrix} =: \mathbf{B} \end{aligned} \quad (3)$$

161 When applied to all time intervals, this resulting feature matrix  $\mathbf{B}$  of respec-  
 162 tive gradients contains directional trend strength indicators for all time series,  
 163 one per row, and with one time interval per column. This representation of time  
 164 interval trends is the basis for the subsequent smoothing process.

One point of concern is the reduction of the dataset due to the given time interval over which a trend is approximated, which requires a sufficiently large set of observations to effectively train artificial neural networks after the reduction (Glorot and Bengio, 2010). Research on the usage of regression derivatives for predictive classification tasks with time series, especially with regard to deep learning approaches, remains sparse. Examples from recent years include the use of such gradients for audio classification via decision trees and support vector machines by Mierswa and Morik (2005), gradients of wavelets for tasks in natural language processing by Gibson et al. (2013) and the addition of such derivatives to dynamic time warping (Grecki and Luczak, 2013).

### 3.2. Infinite impulse response filtering

Clarke et al. (2000) and Gehrig and Menkhoff (2006) state that technical analysis, that is, the prediction of stock price changes based on historical stock market data in the form of time series, is wide-spread in the current investment industry. The exponential moving average is, in this context, one of the dominant techniques used as a lagged indicator for technical analysis. In the general study of time series analysis, it is better known as exponential smoothing (Brown, 1956; Holt, 1957). It serves as a type of infinite impulse response filtering that can be computed recursively as follows, with  $\alpha$  being the smoothing factor used, with  $\alpha \in (0, 1)$  and  $t$  as the time interval indicator:

$$\begin{aligned} s_0 &= x_0 \\ s_t &= \alpha x_t + (1 - \alpha)s_{t-1}, \quad t > 0 \end{aligned} \tag{4}$$

Exponential smoothing is a staple method in digital signal processing, and special consideration has to be placed on the choice of the smoothing factor  $\alpha$ . The sensitivity of a prediction w.r.t.  $s_0$  is inversely correlated with the size of  $\alpha$ , with the average of at least 10 time intervals being widely recommended as the initial value for  $s_0$  (Nahmias, 2009). The applicability of infinite impulse response filters to the smoothing of financial time series has first been emphasized by Genay et al. (2001).

## 4. Empirical validation

### 4.1. Data cleansing and pre-processing

The dataset, which is provided by Thomson Reuters Tick History, features the average stock price per hourly time step from 2011-04-04 to 2016-04-01, covering about five years worth of stock market information for the current 505 S&P 500 stocks, with a combined number of 6,049,849 observations. It contains the price, date and time of an observation and the respective stock's Reuters Instrument Code (RIC).

For an effective use as feature vectors, and to avoid a contamination from a financial perspective, the price series over which the step-wise gradients are computed have to be perfectly aligned and existent for all time steps and used

203 stocks. As the dataset is imperfect, invalid time stamps and non-consistent 204 values for holidays are sorted out, and missing values are reconstructed via the 205 next preceding value of a row with the same RIC. The RICs for which a cut-off 206 value for a minimal number of existent observations is present are then kept. 207 The rest of the stocks, 56 in total, are discarded, as a further reconstruction 208 process would endanger the dataset of becoming corrupted through too many 209 reconstructed insertions.

210 To obtain a perfect alignment despite missing rows, we devised an algorithm  
211 optimized for speed that operates on two combined date-time vectors; one from  
212 an uninterrupted timeline and one per stock matrix, with the dataset being split  
213 into a list of matrices with one list place per RIC.

214 For each list place, the matrix is inflated to avoid higher computational  
215 costs via appending rows. At each discrepancy between the ideal and actual 216 time-  
date vectors, the original matrix within the inflated matrix is shifted one 217 place  
down, the missing row is substituted with the next adjacent values, and 218 the missing  
date-time stamp is inserted. As this process operates on the fine-219 grained level of the  
raw data and only replaces very few missing values, for 220 example for initial public  
offerings after the stock market opens and earlier 221 market closings on Thanksgiving,  
the result on trend features over larger time 222 intervals is negligible.

223 This process is repeated for all list places until all matrices feature a non-  
224 interrupted dataset, after which each matrix is cut at the first occurrence of the 225  
last date-time stamp to deflate the matrix. The prices of the resulting list of 226  
matrices are then extracted and saved as a combined matrix containing only 227 price  
information per column-wise time step, with one stock, as identified by 228 the  
respective RIC, per row of the combined matrix.

229 Subsequently, we compute the simple linear regression from equation (2) on  
230 the values per trading day for each stock, resulting in one price trend approx-231  
imation per trading day. The gradients are then extracted to get the feature 232 matrix  
**B** from equation (3), with 1,241 observations for 449 stocks. The expo-233 nential  
smoothing process from equation (4) with  $\alpha = 0.05$  and the calculation 234 of a  
directional trend change indicator based on a stock price’s trend change rel-235 ative to  
the previous step’s trend, are implemented at the construction for each 236 stock’s  
classification model to avoid storing the correct target values separately.

237 At the end, each matrix is cut at the first occurrence of the last data-time  
238 stamp, and only price information per stock is retained. With this information,  
239 price trend approximations are computed, for each trading day and stock, with  
240 the simple linear regression from equation (2). This leaves us with 1,241  
ob-241 servations for 449 stocks for feature matrix **B** from equation (3), on which  
the 242 exponential smoothing from equation (4) is performed.

#### 243 4.2. Setup of the use case experiments

244 The experimental setup shown in Figure 1 ensures that the experiments  
245 test for time-shifted correlations between time series instead of using a stock’s  
246 own historical information, that is, data of the predicted stock is not part of



the model’s input. This is one of the main differences to related work on time series-based stock market prediction, for example Takeuchi and Lee (2013), with the accuracy used as the metric by which the presence of correlations is measured. Another difference is the use of more short-term, meaning daily instead of monthly, predictions. As we aim to find time-invariant correlations between stocks, five-fold cross-validation with a validation set for early stopping is used to reduce the variability of results, and to use data in a frugal manner (Seni and Elder, 2010).

## INSERT FIGURE 1 HERE (COLOR IN PRINT)

Figure 1: Schematic depiction of the gradient calculation and deep classification model process. For each time series, each time interval’s trend approximation is computed via a simple linear regression over a pre-determined time window, the gradient of which is then extracted to form a time interval’s gradient vector  $(\beta_{1,1}, \beta_{1,2}, \dots, \beta_{1,n})$  as one column of the feature matrix. The latter is then used to train a deep feed-forward neural network for the binary prediction of directional trend changes.

This way, a 60-20-20 split for training, validation and test sets is used for each fold and stock. When performing  $k$ -fold cross-validation, a central concern is the choice of the correct split ratios. The 60-20-20 split is a frequently used rule of thumb in machine learning in which a validation set, in addition to the training and test sets, are necessary. Regarding the larger size of the training set, we want to ensure a large-enough number of data points in the training set to enable the model to learn sufficiently. Conversely, if the validation set is too small, the validation of the learned parameters cannot be properly assessed with a representative sample, while an insufficient size of the test set leads to the final test on unseen data being similarly non-representative. The training examples split from the dataset are normalized element-wise using min-max scaling. Exponential smoothing, as described before, is then applied to the input features of all three sets.

The experiments are run for all time series as a respective target by looping over an index  $i$  for all column-wise stock gradients and splitting the matrix into the target gradients for stock  $i$  and the inputs for the rest of the columns. The time intervals are then shifted one step by clipping the first row of the input matrix and the last value of the output vector. The output vector is subsequently replaced by a binary one-hot representation that indicates whether the gradients for each successive time interval for stock  $i$  are larger or smaller than for the preceding interval.

We repeat these experiments for each stock as the respective target, without past information concerning the stock itself in the inputs. The output vector takes the form of a one-hot representation to indicate whether gradients are larger or smaller than for the preceding interval. Two output nodes are chosen instead of one in accordance with the results of Takeuchi and Lee (2013) and Ding et al. (2015) in favor of this setup, and to ease the structural comparability.

Preliminary test runs, with 20 randomly chosen stocks to avoid cherry-picking hyperparameters for the use case in general, showed the smallest test

set error for 400 nodes per hidden layer, as measured in increments of 50 nodes for up to 800 nodes. The final model used for the experiments features 10 hidden layers, a number that was determined by starting with one hidden layer and, subsequently, adding  $4 - n$  hidden layers for  $n \in \{0, 1, 2, 3\}$ , with the option to add further single layers in case of still-increasing performance. With improved performance up to the addition of nine additional hidden layers, the performance plateaued at this point. Early stopping, together with  $\ell_2$  regularization, is used to prevent overfitting and unnecessary complexity, whereas momentum is applied to prevent stochastic gradient descent from terminating in small-spaced local minima. Dropout, another popular method of preventing overfitting in neural network, is a popular technique in deep learning (Hinton et al., 2012; Srivastava et al., 2014). Initial experiments with different levels of omissions did, however, prevent the model from learning, which can be ascribed to the very high levels of noise in stock market information. Epoch-based learning rate decay is used to find a minimum along the optimizer’s descent path, with learning rate  $\nu$ , decay coefficient  $\delta$  and epoch number  $e$ :

$$\mu_e = \mu_{e-1} \cdot \frac{1}{1 + \delta \cdot e} \quad (5)$$

Hyperbolic tangent functions serve as activation functions, with sigmoid functions at the output layer. The former choice is due to the intent to combat weight saturation, while the latter function is chosen over the softmax function due to the interpretability of the results as independent probabilities, and because these results are not needed to integrate to 1 as inputs for subsequent methods.

The model’s weights are initialized as scaled samples from a zero-mean Gaussian distribution to address the potential of vanishing or exploding gradients, with a variance of  $\frac{2}{n_l}$  and an initial bias of 0, and with  $n_l$  denoting the number of connections in the  $n$ -th layer, allowing for an easy adaptation to future experiments with rectified linear units (Glorot and Bengio, 2010; He et al., 2015).

### 4.3. Accuracy and validation measures

For each stock and fold in each model, a randomly shuffled copy of the predictions is created and tested against the correct targets in addition to the predictions, resulting in mock predictions with a distribution identical to the actual predictions. This copy can be used to test whether the model just learned the distribution of the two output classes in the training set, which would result in very similar accuracies for the actual and mock predictions when compared to the correct targets.

Another case that has to be ruled out is that of a model learning to predict the dominant class of a training set. Two targets are created for each stock and model, each containing only one of the two classes. A model that learns more actionable information from its respective inputs than the dominant class of the training set needs to perform better on the test set than both these one-class mock targets in direct comparison to the correct targets. A fourth validation

metric is computed by taking the maximum of all three metrics' accuracies for each time step, ensuring that a baseline with at least 50% accuracy in each time step is reached in case of distributions that deviate from a 50:50 distribution. As the use of accuracy as the metric of choice can lead to questionable results, we randomly sampled 50 of the dataset's stocks and evaluated them for both accuracy and the area under the curve (AUC) for the receiver operating characteristic, with the latter being a well-established metric for skewed class distributions (Bradley, 1997). While the classes in our work are well-balanced, this exercise provides a validation of the choice of metric, and shines a light on the necessity of taking class distributions into consideration when dealing with classification problems. The accuracy for this subset of stocks is 55.284%, while the AUC score is 0.55278. The necessity of using an additional decimal number demonstrates the similarity of both metrics for well-balanced classes such as the ones used in this paper. Additionally, a linear support vector classifier (SVC) is used on the same gradient-based data to enable a comparison with a simpler approach.

The average accuracy over all stocks is given as the standard method to assess the predictive power of a model. These accuracies do, however, not indicate whether the predictions' variations are too large to be considered successful, that is, whether the volatility of the model grows too large. For this reason, we also provide accuracies for the three baseline mock predictions. In addition, the  $p$ -values for the predictions' accuracies via an upper-tail test are calculated for each of the baselines and an additional baseline that contains the highest accuracy among the three baselines for each stock, that is, for each model. The null hypothesis  $H_0$  in each case is that the predictions' accuracies are not significantly larger than the respective baseline, with a very strict significance level of  $\alpha = 0.001$ .

Lastly, notched box plots are a commonly used visualization tool for descriptive statistics, using the respective data's quartiles to allow for an intuitive representation. Non-overlapping notches for two boxes indicate a statistically significant median difference at 95% confidence. Welch's  $t$ -test is used to achieve higher reliability for unequal variances.

In order to rule out the simpler explanation that only a few observations are sufficient to identify general market behavior, and to address the possibility that using only past data to predict future trend changes yields no better accuracies than cross-validation that also takes future observations into account, an additional experiment is conducted: For 20 randomly sampled non-repeating stocks, and with  $N$  being the size of the whole dataset, a neural network model is first fully trained, as for the primary experiments, on the first observation. Then, for  $l = 2$  to  $(N - 1)$ , the model is iteratively updated by training for 3 epochs on observations 1 to  $l$  as the training set, while being tested on observations  $l + 1$  to  $N$  as the test set. This procedure can be visualized as sliding a divisive line along across the dataset, training on an ever-increasing training set while simultaneously reducing the test set. The last point has to be kept in mind, as a small test set does not offer a good representation of the data. For this reason, the described procedure is stopped at the point when only 10% of observations

371 remain in the test set to still deliver a viable estimate.

#### 372 4.4. Results of the experiments

373 The use of  $p$ -values has to be viewed with caution, as criticism of their often  
 374 incorrect use has risen in recent years. In 2016, the American Statistical Associ-375  
 ation published an official warning regarding the wide-spread misuse of  $p$ -values 376  
 (Wasserstein and Lazar, 2016). Accordingly, the  $p$ -values are given in combi-377 nation  
 with other metrics such as the lower boundary for differences in means 378 given a  
 99.9% confidence interval, notched box plots for median differences and 379 quartile  
 distributions, and accuracies for models and baselines.

380 Figure 2 shows the accuracy of 58.10% listed in Table 1 significantly above  
 381 all baselines, both for the means as measured by the  $p$ -values and the medians as 382  
 indicated by the box plots, with neither the notches nor the boxes overlapping. 383 The  
 first and third quartiles are, however, spread wider for the model when 384 compared to  
 the baselines, with the exception of the SVC. In Table 1, the 385 accuracies for the  
 model, the different baselines and a simple linear support 386 vector classifier are given,  
 as well as the  $p$ -value results with regard to the  
 means and the minimal difference for a 99.9% confidence interval.

### INSERT FIGURE 2 HERE (COLOR IN PRINT)

Figure 2: Statistical validation and accuracies. Part *a* shows a notched box plot representation  
 of the results. Non-overlapping notches indicate a statistically significant difference in medians  
 at 95% confidence, which is the case for all baselines. Here, *class 1* is the forecast that stock  
 trends will change downwards, while *class 2* refers to an upward change. *predictions* denotes the  
 performance of the model, *svc* shows the boxplot for the support vector classifier (SCV), *random*  
 is the performance of a random forecast with the same distribution as the actual predictions,  
 and *best-of* represents the best result among the baselines for each time step. Part *b* shows an  
 intuitive visualization of the mean accuracies. The upper blue line represents the model’s  
 predictions, whereas predicting solely upward changes or downward changes is drawn in cyan  
 and green, respectively. Red denotes the random vector with the same class distribution as the  
 model’s predictions. The upper blue line represents the model’s predictions and the red line the  
 SVC’s predictions, whereas predicting solely upward changes or downward changes is drawn in  
 light blue and green, respectively. Fulvous denotes the random vector with the same class  
 distribution as the model’s predictions. The averages for the model results and the randomized  
 predictions are drawn as lines of the same color.

387  
 388 While the focus of this work is the existence and exploitation of lagged  
 389 correlations, one might be left to wonder what impact the omission of target  
 390 stock information in the inputs of each stock’s run has on the results. To answer  
 391 this question, we repeat the experiment described above, with this information  
 392 included in the inputs. The reported result of 58.10% without this information  
 393 increases to 58.54% when information about the target stock is included. While  
 394 this shows that a stock’s past behavior provides additional information, this  
 395 result demonstrates the model’s ability to infer most of the relevant information  
 396 from lagged correlations between the target stock and other stocks.

397 As described in the discussion of accuracy and validation measures, a feed-  
 398 forward neural network is trained on the first observation and then re-trained 399 for  
 3 epochs on the dataset extended by the previous time step for the remaining

accuracies of predictions					
model	rand.	class 1	class 2	best-of	SVC
0.5810	0.5002	0.4955	0.5045	0.5092	0.5474
variances of accuracies					
model	rand.	class 1	class 2	best-of	SVC
$6.12e^{-4}$	$1.43e^{-4}$	$5.93e^{-5}$	$5.93e^{-5}$	$4.50e^{-5}$	$10.87e^{-4}$
tests against baselines					
	rand.	class 1	class 2	best-of	SVC
<i>p</i> -value	$< 1e^{-3}$	$< 1e^{-3}$	$< 1e^{-3}$	$< 1e^{-3}$	$< 1e^{-3}$
min. diff.	0.0767	0.0816	0.0727	0.0679	0.0276

Table 1: Accuracies, variances and baseline comparison. Accuracies and variances for the model’s predictions (*model*), as well as for a best-of of all baselines (*best-of*), the randomized baseline with the same class distribution (*rand.*), comparative results for a support vector classifier (*SVC*) and results for predictions for one class exclusively are provided. *class 1* denotes upwards stock trend changes and *class 2* downward stock trend changes. *p*-values and minimum differences are provided for comparison against the chosen baselines.

time steps’ prediction. This process is repeated for 20 randomly chosen stocks, after which the results for the predictions are averaged. This process is implemented until 90% of the dataset is used as the training set in order to allow for a small number of observations to remain as a test set towards the end. Since the robustness of the obtained results is an important factor to be taken into account, leaving a sufficient amount of time steps as a test is crucial. If the process of shifting from the test to the training set would be taken to the extreme by continuing until only one stock is left in the test set, the results would be based on a non-representative number of time steps. From a financial markets perspective on robustness, the setup of this experiment allows for the assessment of the model’s performance over multiple years, and thus across changing market conditions over time. For a measurement of the model’s accuracy during the end of the experiments, the average of the last 100 time steps is taken and results in an average accuracy of 60.23%.

Notably, this result outperforms the previous experiment that uses cross-validation, which indicates an advantage of learning exclusively from data before the respective test example that is to be classified instead of using training examples from both before and after the test example to extract time-invariant market dynamics. Within the framework of the stock market, an explanation for this observation is that lagged trend correlations in financial markets feature information that can be used for predictions about the future in which the lagged effect takes place, but to a lesser degree for predictions about past observations. The learned information is, in this case, local within time, render-

ing cross-validation less effective when compared to training on past data with online updates to adapt to new information. The accuracies of this additional experiment over the course of the training process are depicted in parts  $a_1$  and  $b_1$  of Figure 3.

To determine whether the initial sharp increase in average accuracy and the subsequent slower and quasi-linear increase are a result of the the market or the learning, we repeat the same experiment without the first 100 time steps in which the sharp increase takes place, which means starting at a later point in time and excluding the model from learning about the training examples before that. Part  $a_1$  of Figure 3 shows the initial rise in accuracy for this cut-down dataset following a steeper path at first, but reaching the full dataset’s first elbow accuracy later in relation to the number of steps since starting the training process. It subsequently follows a more concave curve instead of the full dataset’s quasi-linear increase in accuracy during that phase. Similarly, the final level of the cut-down dataset is reached at about the same time as the full dataset’s corresponding accuracy, again in relation to the number of steps the model is trained on instead of the actual point in time due to the omission of the first 100 time steps and, therefore, a later point in real time.

The resulting average prediction accuracy is 59.50% due to the full dataset’s later and more pronounced accuracy peak, although both experiments feature the same dip around time step 900, or time step 1000 for the full dataset. While reaching the peaks in parts  $a_1$  and  $b_1$  of Figure 3 seems to be a result of the number of time steps the models are trained on, the dips at 900 and 1000, respectively, seem to be a result of the market at that time. These results suggests that the primary features of the time series are due to the market, whereas the profiles of the first initial accuracy increase and the subsequent rise, as well as the lower maximum towards the end, can be attributed to the lack of the information contained in the deleted first 100 training examples, inhibiting the portion of long-term market information that would otherwise be extractable from the latter.

### INSERT FIGURE 3 HERE (COLOR IN PRINT)

Figure 3: Averaged and individual accuracies for the experiment with omitted data for the first 100 time steps. For the same 20 stocks selected via random sampling, a model is iteratively updated for 3 epochs and with the last time step’s information as an extension of the training set after each new time step. Parts  $a_1$  and  $a_2$  show the results for averaged and individual accuracies for the case without the first 100 time steps, respectively, whereas parts  $b_1$  and  $b_2$  show the same visualizations for the full dataset containing the first 100 time steps.

Another result of the omission of the first 100 time steps can be seen when comparing the individual accuracies for the targeted stocks in parts  $b_1$  and  $b_2$  of Figure 3. While the individual stocks in both datasets’ accuracies feature similar evolutions, the full dataset’s accuracies show a clearer trend towards higher accuracies. This observation reflects the more linear ascend of the average accuracy and can be viewed as an indicator of market behavior information from multiple years into the past still influencing the market behavior’s overall

predictability in the present, the implications of which are discussed later.

Figure 4 is comprised of heatmaps for both cases in order to get a better overview of the individual 20 stocks featured in Figure 3. As can be seen, the development of the predictive accuracy for individual stocks follows similar progressions for both experiments, albeit with slightly different distributions and starting points of high-accuracy periods. A more general finding in Figure 4 is that some stocks are considerably easier to predict solely on other stocks' prior behavior than others.

## INSERT FIGURE 4 HERE (COLOR IN PRINT)

Figure 4: Heatmaps for the experiment with omitted data for the first 100 time steps. For the same 20 stocks selected via random sampling, a model is iteratively updated for 3 epochs and with the last time step's information as an extension of the training set after each new time step. Part *a* shows the results for averaged and individual accuracies for the case without the first 100 time steps, whereas part *b* show the results for the full dataset.

## 5. Discussion

### 5.1. Relevance for trend

The hypothesis about general time series analysis via such network model prediction is reinforced by the experiments: The evidence strongly suggests that deep feed-forward neural networks can be used to consistently learn and, for previously unseen data, act with an accuracy above predetermined baselines on time-shifted correlations of gradients that are computed step-wise for complex time series, with only the previous interval of other series instead of the target one as input features. While adding information about the target stock in the inputs is shown to provide only marginal increases in performance, a simple linear SVC baseline performs significantly above naïve baselines, further bolstering the relevance of the feature engineering used in this work. The approach of this paper could be applied to other forecasting problems that involve non-linear interactions between a large number of noisy time series and lagged effects of their respective trend behavior, for example the metrics in areas as diverse as consumer behavior and epidemic dynamics of infectious diseases. For practitioners relying on expert systems to inform decisions about the future in noise-laden environments, breaking through that noise is a common issue. As a novel use and application of gradient-based feature engineering, in combination with smoothing techniques described in Section 3.2, this paper delivers general evidence for the viability as an expert system in highly noisy dynamical systems subject to time series prediction problems.

### 5.2. Comparison to related research

Both Takeuchi and Lee (2013) and Batres-Estrada (2015) use deep learning models for the binary month-wise trend prediction of target stocks, based on historical stock market data of the preceding 12 months, with resulting accuracies of 53.36% and 52.89%. We want to emphasize that these papers don't

work on the same time intervals and use additional features instead of just past stock prices, which means that a comparison should be taken with a grain of salt. These results are however, the closest available comparison of feed-forward deep learning models for trend prediction based on historical stock market data. When directly comparing the accuracies, our model outperforms both by a noticeable margin, with 58.10% for cross-validation and 60.23% for training exclusively on past observations and re-training the model for a few iterations before each new prediction. The same, although to a lesser degree, holds true for the simple linear SVC used as an additional baseline, indicating the viability of the proposed feature engineering.

### 5.3. Implications for financial economics

Firstly, the findings add to the existing evidence against the random walk hypothesis as popularized by Malkiel (1973) and others, that is, the notion that stock prices follow random walks with inherently unpredictable behavior. Our findings in this respect are consistent with a growing view in the financial economics literature. Consequently, the results also challenge the related efficient-market hypothesis developed by Fama (1965), which has since become a staple in the field of financial economics. However, in order to convincingly argue for an empirical rebuttal of the efficient-market hypothesis, a simulation taking trading costs into account may be necessary, which could be a topic for further research and would have to show consistent outperformance in the face of these costs. Lastly, the higher accuracy for models trained exclusively on past observations and with online updates have implications for the interpretation of the underlying structures learned by the models: The latter appear to partially adapt to changing correlations between stocks, meaning that the learned information is, to a degree, temporally constrained instead of reflecting only general market behavior. This shows that stock markets are, over periods of multiple years, non-stationary in their correlated behavior even in the case of summary findings for the S&P 500.

Another relevant finding is that the inclusion of information about the respective target stock does provide additional information that improves predictive results, but that this increase is rather small and the model is able to infer most relevant information from lagged correlations with other stocks. This recovery of most of the information relevant to the predictions in this paper further strengthens the arguments against both the random walk hypothesis and the efficient-market hypothesis in most of its forms.

The efficient-market hypothesis allows for three forms of information-based market efficiency. Our results could be consistent for a constrained version of the most basic form, the weak-form market efficiency. Specifically, this would be for a case where the hypothesis allows for prediction methods that reliably outperform the market and can only be implemented by a sufficiently small number of investors so as not to result in a new aggregate equilibrium for a typical modern economy. With a negligible amount of capital involved in the context of the whole market, some agents, such as select quantitative hedge



539 funds or individuals, could consistently realize above-average returns, thus re-540  
 ducing the weak form efficient-market hypothesis to a context-based version. A 541  
 time-specific weak form efficient-market hypothesis would, in effect, acknowl-542 edge  
 that the hypothesis does not apply to the overall market, but does for a 543 majority of  
 the trading stakeholders due to restrictions regarding the method-544 ology and the  
 capital involved in deploying such strategies. This view bears 545 argumentative  
 resemblance to the adaptive-market hypothesis by Lo (2004), 546 perceiving the  
 efficient-market hypothesis as not necessarily incorrect, but in-547 complete, resulting  
 in an attempt to merge the efficient-market hypothesis with 548 behavioral economics  
 by applying principles of biological evolution. For ex-549 ample, Lo (2004) states that a  
 high competition for scarce resources leads to 550 highly efficient markets, whereas a  
 competition for abundant resources among 551 few "species" in a financial market  
 diminishes overall market efficiency.

#### 552 5.4. Further research suggestions

553 High frequency trading (HFT) is the use of high-frequency stock market  
 554 data characterized by short holding times and high rates of cancellation for 555  
 equities and futures trading in a fully automated manner (Menkveld, 2013). 556 It  
 remains a driving force in stock markets, with double-digit shares of total 557 trading  
 volumes across different markets and competition mostly between such 558 HFT  
 algorithms. Using our model, the interaction between those algorithms in 559 the form  
 of lagged correlations could be further investigated to better explain 560 the behavior of  
 this part of the stock market. Another approach is the use of 561 wavelets, that is, the  
 results of time-frequency transformation to compute a local 562 variations  
 representation on different scales suitable to combat noise (Nason and 563 von Sachs,  
 1999).

564 As described earlier, gradient-based wavelet approaches are used in the areas  
 565 of natural language processing and acoustic classification, and our model could 566 be  
 used with wavelets as a more elaborate way to extract relevant information 567 from  
 intervals in time series. The question remains whether an increased so-568 phistication  
 equals a better model performance. While the linear SVC provides 569 a simple off-the-  
 shelf baseline, serving the same purpose as logistic regression in 570 similar papers, it  
 still performs significantly better than naïve baselines. As a 571 result, further fine  
 tuning and modification of SVCs provide a viable approach in 572 problems where  
 training speed is more important than accuracy maximization.

## 573 6. Conclusion

574 In conclusion, we have shown in this paper that the results with regard 575 to the  
 investigated hypothesis are positive under conscientious observance of 576 statistical  
 validation measures. The results of the experiments deliver evidence 577 for the  
 viability of a combination of deep feed-forward neural networks and 578 exponential  
 smoothing applied to gradient-based features for directional trend 579 change  
 predictions with non-linear correlations of large numbers of noisy time 580 series in the  
 form of historical stock market data. More generally, our findings

581 demonstrate the value of deep learning approaches to time series analysis and  
 582 show that linear regression derivatives provide useful features to extract such  
 583 complex interdependencies, offering a simple indicator of trends with a high  
 584 predictive value.

585 The findings in this paper also have implications for modern finance theory,  
 586 delivering strong evidence against both the random walk and efficient-market hy-  
 587 potheses. The postulations of the efficient-market hypothesis may, however, be  
 588 adapted to allow for the findings as presented here. While indeed all three forms  
 589 of the efficient-market hypothesis are inconsistent with the evidence, tweaking  
 590 the weak form of the efficient-market hypothesis could lead to consistency with  
 591 our findings. Furthermore, while the presented findings are successfully tested  
 592 on stock market data and have interesting implications for hypotheses within  
 593 financial economics, they should also be applicable to other fields dealing with  
 594 trend forecasts in time series. In addition, as simple arbitrage approaches to in-  
 595 vestment became less effective due to the growing use of such methods over the  
 596 last decades, it remains to be seen whether deep learning methods such as the  
 597 one we discuss in this paper will see a similar spread and, consequently, a failure  
 598 to perform due to a large enough number of market participants operating with  
 599 related techniques.

## 600 References

- 601 Baggenstoss, P. M. (2015). Derivative-augmented features as a dynamic model  
 602 for time-series. In *Proceedings of the 23rd European Signal Processing Con-*  
 603 *ference (EUSIPCO)*, pages 958–962.
- 604 Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.  
 605 Master’s thesis, KTH Royal Institute of Technology.
- 606 Boyacioglu, M. A. and Avci, D. (2010). An adaptive network-based fuzzy infer-  
 607 ence system (anfis) for the prediction of stock market return: The case of the  
 608 istanbul stock exchange. *Expert Systems with Applications*, 37(12):7908–7912.
- 609 Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation  
 610 of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.
- 611 Brown, R. G. (1956). *Exponential smoothing for predicting demand*. Cambridge,  
 612 Massachusetts: Arthur D. Little Inc.
- 613 Cao, L. J. and Tay, F. E. H. (2003). Support vector machine with adaptive  
 614 parameters in financial time series forecasting. *IEEE Transactions on Neural*  
 615 *Networks and Learning Systems*, 14(6):1506–1518.
- 616 Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., and Oliveira,  
 617 A. L. (2016). Computational intelligence and financial markets: A survey and  
 618 future directions. *Expert Systems with Applications*, 55:194–211.

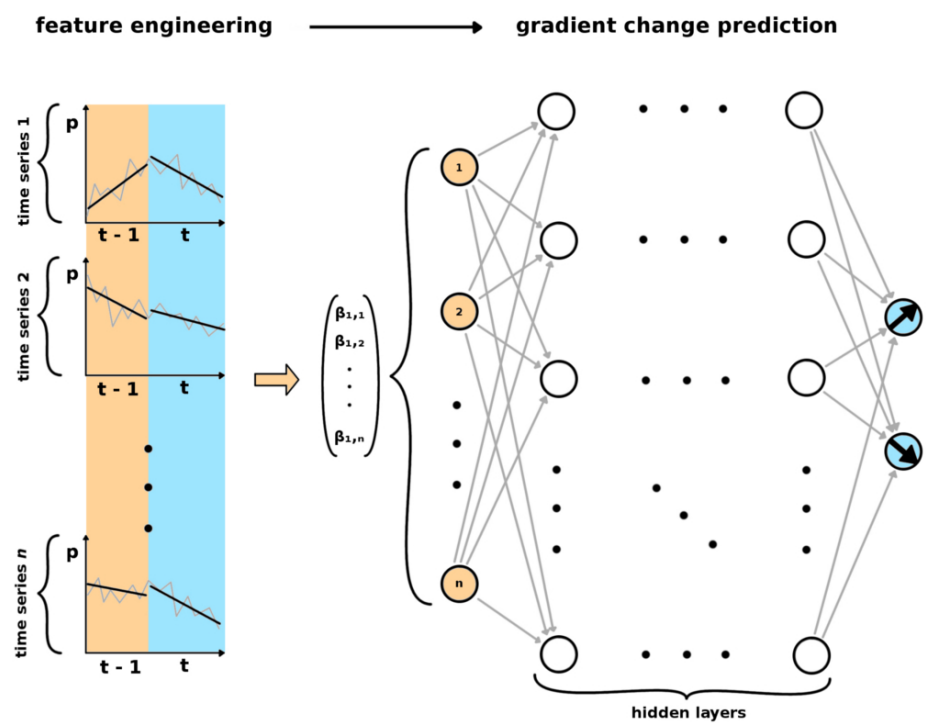
- 619 Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., and Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, 112:353–371.
- 623 Chen, N.-F., Roll, R., and Ross, S. A. (1986). Economic forces and the stock market. *The Journal of Business*, 59(3):383–403.
- 625 Clarke, J. P., Jandik, T., and Mandelker, G. (2000). The efficient markets hypothesis. In *Expert Financial Planning: Advice from Industry Leaders*, pages 126–141.
- 628 Cootner, P. (1964). *The random character of stock market prices*. M.I.T. Press.
- 629 Darrat, A. F. and Zhong, M. (2000). On testing the random-walk hypothesis: A model-comparison approach. *Financial Review*, 35(3):105–124.
- 631 Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI’15)*, pages 2327–2333.
- 634 Doran, J. S., Peterson, D. R., and Wright, C. (2010). Confidence, opinions of market efficiency, and investment behavior of finance professors. *Journal of Financial Markets*, 13(1):174–195.
- 637 Drakos, K. (2004). Terrorism-induced structural shifts in financial risk: airline stocks in the aftermath of the september 11th terror attacks. *European Journal of Political Economy*, 20(2):435–446.
- 640 Fama, E. F. (1965). The behaviour of stock market prices. *Journal of Business*, 38(1):34–105.
- 642 Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.
- 644 Fama, E. F. and French, K. R. (1988). Permanent and temporary components of stock prices. *Journal of Political Economy*, 96(2):246–273.
- 646 Gehrig, T. and Menkhoff, L. (2006). Extended evidence on the use of technical analysis in foreign exchange. *International Journal of Finance & Economics*, 11(4):327–338.
- 649 Genay, R., Genay, R., Seluk, F., and Whitcher, B. J. (2001). *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Amsterdam: Elsevier, 1 edition.
- 652 Gibson, J., Segbroeck, M. V., Ortega, A., Georgiou, P. G., and Narayanan, S. (2013). Spectro-temporal directional derivative features for automatic speech recognition. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013)*, pages 872–875.

- 656 Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training  
657 deep feedforward neural networks. In Teh, Y. W. and Titterton, M., edi-  
658 tors, *Proceedings of the Thirteenth International Conference on Artificial In-*  
659 *telligence and Statistics (AISTATS10)*, volume 9 of *Proceedings of Machine*  
660 *Learning Research*, pages 249–256. PMLR.
- 661 Grecki, T. and Luczak, M. (2013). Using derivatives in time series classification.  
662 *Data Mining and Knowledge Discovery*, 26(2):310–331.
- 663 He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Sur-  
664 passing human-level performance on imagenet classification. In *Proceedings of*  
665 *the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages  
666 1026–1034.
- 667 Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of  
668 data with neural networks. *Science*, 313(5786):504–507.
- 669 Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov,  
670 R. (2012). Improving neural networks by preventing co-adaptation of feature  
671 detectors. *Computing Research Repository*, abs/1207.0580.
- 672 Ho, T. S. Y. and Stoll, H. R. (1983). The dynamics of dealer markets under  
673 competition. *The Journal of Finance*, 38(4):1053–1074.
- 674 Holt, C. C. (1957). Forecasting trends and seasonal by exponentially weighted  
675 averages. *Office of Naval Research memorandum*, (52).
- 676 Kendall, M. G. and Hill, A. B. (1953). The analysis of economic time-series-  
677 part i: Prices. *Journal of the Royal Statistical Society. Series A (General)*,  
678 116(1):11–34.
- 679 Lin, X., Yang, Z., and Song, Y. (2009). Short-term stock price prediction based  
680 on echo state networks. *Expert Systems with Applications*, 36(1):7313–7317.
- 681 Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an  
682 evolutionary perspective. *Journal of Portfolio Management*, 30(1):15–9.
- 683 Lo, A. W. and MacKinlay, A. C. (1987). Stock market prices do not follow  
684 random walks: Evidence from a simple specification test. Working Paper  
685 2168, National Bureau of Economic Research.
- 686 Malagrino, L. S., Roman, N. T., and Monteiro, A. M. (2018). Forecasting stock  
687 market index daily direction: A bayesian network approach. *Expert Systems*  
688 *with Applications*, 105:11–22.
- 689 Malkiel, B. G. (1973). *A random walk down Wall Street*. New York: W. W.  
690 Norton & Company, Inc.
- 691 McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots.  
692 *The American Statistician*, 32(1):12–16.

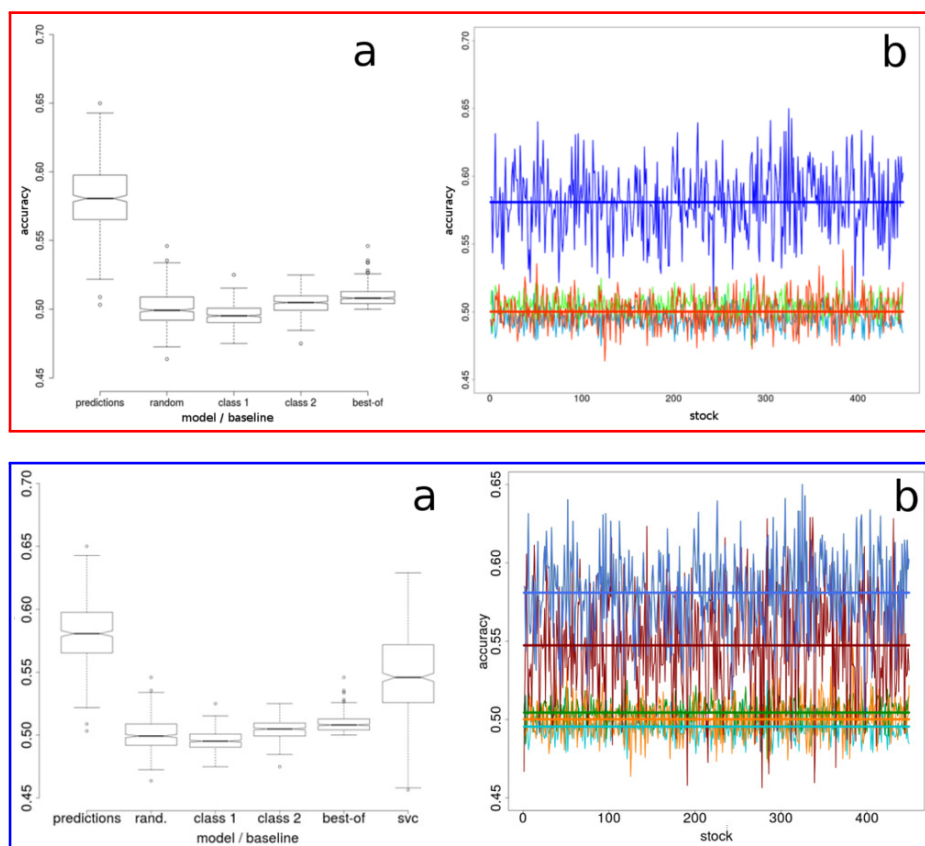
- 693 Menkveld, A. J. (2013). High frequency trading and the new market makers.  
694 *Journal of Financial Markets*, 16(4):712–740.
- 695 Mierswa, I. and Morik, K. (2005). Automatic feature extraction for classifying  
696 audio data. *Machine Learning*, 58(2):127–149.
- 697 Nahmias, S. (2009). *Production and operations analysis*. Boston and New York:  
698 McGrawHill/Irwin Series in Operations and Decision Sciences.
- 699 Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., and  
700 Muharemagic, R. W. E. (2015). Deep learning applications and challenges  
701 in big data analytics. *Journal of Big Data*, 2(1):1–21.
- 702 Nason, G. P. and von Sachs, R. (1999). Wavelets in time series analysis. *Catholic*  
703 *University of Louvain - Institute of Statistics Paper Series*, (9901).
- 704 Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., and Ngo, D. C. (2014). Text  
705 mining for market prediction: A systematic review. *Expert Systems with*  
706 *Applications*, 41(16):7653–7670.
- 707 Nesreen, A., Atiya, A., Gayar, N. E., and El-Shishiny, H. (2010). An empricial  
708 comparison of machine learning models for time series forecasting. *Economet-*  
709 *ric Reviews*, 29(5-6):594–621.
- 710 Ng, A. (2012). Machine learning and ai via brain simulations. In *Lectures of the*  
711 *26th Annual Conference on Neural Information Processing Systems (NIPS*  
712 *2012)*.
- 713 Perryman, A. A., Butler, F. C., Martin, J. A., and Ferris, G. R. (2010). When  
714 the ceo is ill: Keeping quiet or going public? *Business Horizons*, 53(1):21–29.
- 715 Saad, E. W., Prokhorov, D. V., and Wunsch, D. C. (1998). Comparative study  
716 of stock trend prediction using time delay, recurrent and probabilistic neural  
717 networks. *IEEE Transactions on Neural Networks*, 9(6):1456–1470.
- 718 Seni, G. and Elder, J. (2010). *Ensemble Methods in Data Mining: Improving*  
719 *Accuracy Through Combining Predictions*. San Rafael, California: Morgan &  
720 Claypool.
- 721 Sitte, R. and Sitte, J. (2002). Neural networks approach to the random walk  
722 dilemma of financial time series. *Applied Intelligence*, 16(3):163–171.
- 723 Skabar, A. and Cloete, I. (2002). Neural networks, financial trading and the effi-  
724 cient markets hypothesis. In *Proceedings of the 24th Australasian Conference*  
725 *on Computer Science*, pages 241–249.
- 726 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.  
727 (2014). Dropout: A simple way to prevent neural networks from overfitting.  
728 *Journal of Machine Learning Research*, 15:1929–1958.

- 729 Summers, L. H. (1986). Does the stock market rationally reflect fundamental  
730 values? *The Journal of Finance*, 41(3):591–601.
- 731 Takeuchi, L. and Lee, Y.-Y. A. (2013). Applying deep learning to enhance  
732 momentum trading strategies in stocks. working paper.
- 733 Wasserstein, R. L. and Lazar, N. A. (2016). The asa’s statement on p-values:  
734 Context, process, and purpose. *The American Statistician*, 70(2):129–133.
- 735 Weng, B., Lu, L., Wang, X., Megahed, F. M., and Martinez, W. (2018). Predict-  
736 ing short-term stock prices using ensemble methods and online data sources.  
737 *Expert Systems with Applications*, 112:258–273.
- 738 White, H. (1988). Economic prediction using neural networks: the case of ibm  
739 daily stock returns. *Proceedings of the IEEE 1988 International Conference*  
740 *on Neural Networks*, 2:451–458.
- 741 Zhang, G., Patuwo, B., and Hu, M. Y. (1997). Forecasting with artificial neural  
742 networks: The state of the art. *International Journal of Forecasting*, 14(1):35–  
743 62.
- 744 Zhang, G. P. and Qi, M. (2005). Neural network forecasting for seasonal and  
745 trend time series. *European Journal of Operational Research*, 160(2):501–514.
- 746 Zhang, J., Cui, S., Xu, Y., Li, Q., and Li, T. (2018). A novel data-driven stock  
747 price trend prediction system. *Expert Systems with Applications*, 97:60–69.
- 748 Zhang, Y. and Wu, L. (2009). Stock market prediction of s&p 500 via combina-  
749 tion of improved bco approach and bp neural network. *Expert Systems with*  
750 *Applications*, 36(5):8849–8854.

751 Figure 1

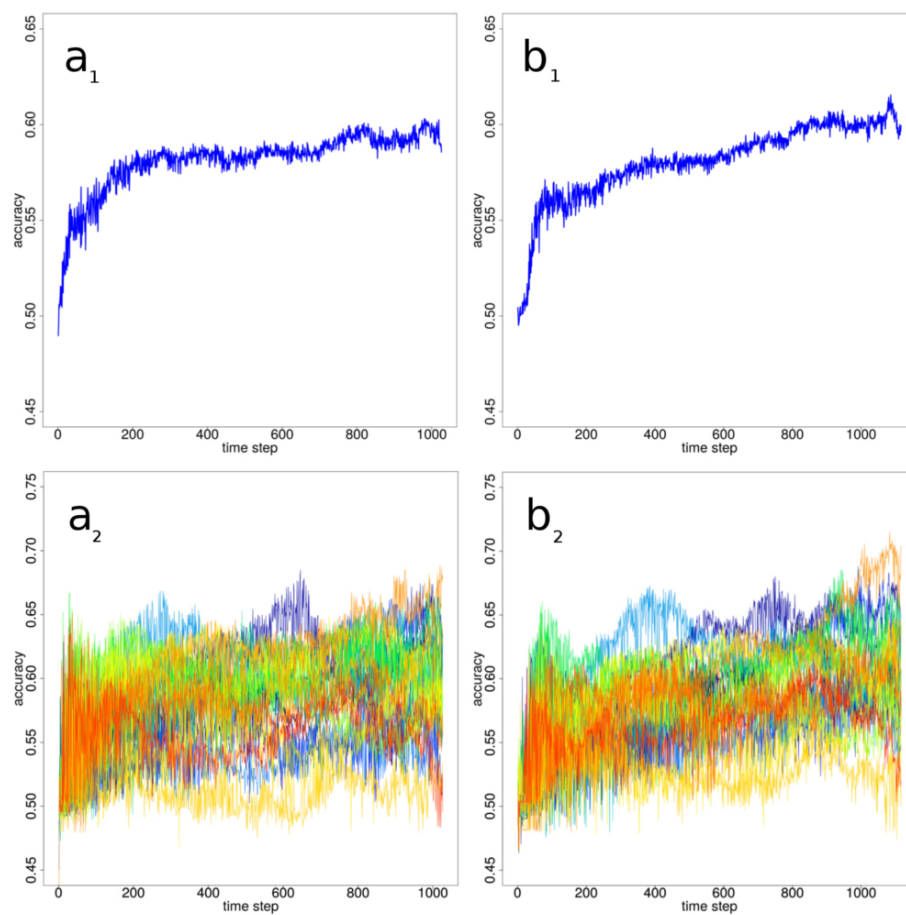


752 **Figure 2**





753 **Figure 3**



754 **Figure 4**

